C8

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/850,183	05/07/2001	Mark A. Kampe	80168-0106	9753

32658 7590 01/14/2005

HOGAN & HARTSON LLP
ONE TABOR CENTER, SUITE 1500
1200 SEVENTEEN ST.
DENVER, CO 80202

EXAMINER	
PROCTOR, JASON SCOTT	
ART UNIT	**PAPER NUMBER**
2123	

DATE MAILED: 01/14/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/850,183	KAMPE, MARK A.
	Examiner	Art Unit
	Jason Proctor	2123

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 November 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *07 May 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-19 have been presented for reconsideration in view of Applicant's amended claim language and arguments.

Response to Arguments

2. The arguments submitted by the Applicant on November 10, 2004 have been fully considered. The Examiner's response is as follows.

3. Regarding Applicant's response to the Examiner's objections to the drawings: The Examiner respectfully apologizes for not clarifying the deficiencies of the drawings submitted on May 7, 2001. The Examiner notes that Fig. 5 has hand written reference characters and is therefore informal. Correction with formal drawings is respectfully requested.

4. Regarding Applicant's corrections to the specification: The Examiner thanks the Applicant for correcting the minor spelling errors in the specification.

35 U.S.C. § 112

5. Regarding Applicant's response to the 35 U.S.C. § 112 rejections of claims 8 and 18, Applicant argued:

> The terms considered indefinite in claim 8 are believed to have accepted meaning in the technical arts and additionally definitions or exemplary definitions are provided In Applicant's specification. Therefore, additional defining language is not believed necessary to meet the requirements of § 112, second paragraph. The other claim rejections are believed addressed by the claim amendments presented herein.

6. The Examiner respectfully asserts that after a careful review of Applicant's specification that it is still unclear exactly what are the "metes and bounds" of Applicant's current claim language. Further, in regard to the terms "warm recoverable errors" and "non-warm recoverable errors", a search of relevant technical literature has failed to disclose an accepted meaning for Applicant's use of these terms. A review of Applicant's specification merely discloses the terms being used to describe processes as opposed to precise definitions. The Examiner notes that page 24 of the specification states:

> Thus, software availability may be impacted by errors that result in recovery actions in the applications, or warm recoverable, or errors that result in recovery actions on the node or cluster, or non-warm recoverable.

Page 7 of the specification states:

> Operating system 104 and software application 108 can be considered the software components of node 102. Repairs to software components may include restarting the application, rebooting node 102, and other activities that should not necessitate hardware fixes or repairs.

7. These exemplary uses of the terms fail to unambiguously define the their meaning. For example, the Examiner is confused as to whether rebooting the node is to be construed as a repair to a software component and therefore a warm recoverable error, or if rebooting the node is to be construed as a recovery action on the node and therefore a non-warm recoverable error.

8. Regarding Applicant's response to the 35 U.S.C. § 112 rejections of claims 1, 4, 8, 9, 11, and 17, the Examiner notes that instant amendments to Applicant's claims have overcome some basis for the earlier rejection. The Examiner withdraws the earlier 35 U.S.C. § 112, 2nd paragraph rejections of claims 1, 4, 8, 9, 11, and 17.

9. Because Applicant has failed to directly refute the Examiner's claim interpretations in regards to the terms warm recoverable error and non-warm recoverable error, the Examiner asserts that those terms have the meaning as put forth in the previous action's claim interpretation.

10. Further, it is noted by the Examiner that due to the fact that Applicant has failed to provide a clear and concise definition of the terms warm recoverable error and non-warm recoverable error, the Examiner asserts that Applicant's specification is not enabling for these terms in that one of ordinary skill in the art, at the time the invention was made would not be able to make and/or use Applicant's invention because of a lack of clear and concise definition of these terms. (Please see 35 U.S.C. §112 1st rejection prompted by Applicant's lack of a clear and concise definition.) For the purposes of compact prosecution, the Examiner has provided claim interpretation of the terms warm recoverable error and non-warm recoverable error.

11. Further, Applicant has failed to clarify the term "fraction of recovery failures" and related steps. Therefore the previous rejections of claims 9, 10, 11, and 17 that rely on this term and related steps is maintained. While the Examiner appreciates that Applicant has amended the claim in response to the 35 U.S.C. § 112, second paragraph rejections regarding the phrase "determining a fraction of recovery failures", no clarification has been provided regarding the relative term "a fraction of recovery failures". The Examiner observes that both zero and one are fractions that could correspond to no attempts resulting in failures and all attempts resulting in failures, respectively. Since the claim limitations merely recite determining the fraction and

make no direct use of the result, the broadest reasonable interpretation of these claims would include an invention where none or all of the recovery attempts fail, in which case the result of the claimed step (a fraction of recovery failures) would be determined without performing the recited algebraic step.

12. Therefore it is unclear if the algebraic step is a required component of the invention; equivalently, it ambiguous whether some of the recovery attempts must fail in order to teach the claimed invention.

13. Further, it is undefined how the claimed invention would perform when there have been no attempted recoveries from said warm recoverable software errors. As recited, the invention appears to divide by zero, making the details of its operation vague and indefinite.

35 U.S.C. § 102

14. Regarding Applicant's response to the rejections under 35 U.S.C. § 102 of claims 1-5, Applicant argued:

> Regarding claim 1, the Office Action cites Zager at col. 11, lines 10-16 for teaching "the model includes an aggregate failure rate and aggregate repair time for each said classes of failures in the form of aggregate fault and impact data." Applicant disagrees with this interpretation of Zager. Zager at this citation and elsewhere provides no discussion of determining an aggregate failure rate or an aggregate repair time for various classes of failures of a software component.

15. The Examiner respectfully traverses the Applicant's arguments and upholds the earlier 35 U.S.C. § 102 rejections of claims 1-5 based on the following remarks.

16. Regarding claim 1, at column 11, lines 19-23, Zager teaches:

> Any detected change of an individual managed object's state to some undesirable value is a "fault". (A MO's state is simply the set of values that its various attributes have.) An event is a representation of a fault in the model, or of the inverse, a recovery from a fault.

17. Zager clearly teaches that objects experience faults. A failure constitutes a change in state to some undesirable value and is therefore taught by the concept of a fault. Zager clearly teaches that recoveries from faults are modeled in the system. A recovery from a fault is functionally equivalent to a repair in response to a failure.

18. At column 12, lines 62- 65, Zager teaches:

> 5. Impact/root cause analysis: Determine the systemic, or contextual, significance of condition-changing event(s) (that is, determine the effect on the condition of the overall system that results from the specific event).

19. At column 14, lines 53-65, Zager teaches:

> Suppose that a disk drive connected to a computer node fails in the external system. Applications depending on the drive will stop functioning also. Both the disk drive and (probably) the applications will emit error messages. In the model of the preferred embodiment, the disk drive MO receives a message indicating an event that is inherently a root-cause event (the disk drive failure), and emits a state change message to its dependents, including the MO's for the applications in question. The application MO's consequently change their state. In this case, the model is predicting that the applications will feel the impact of the disk drive failure, and the invention labels the application MO state changes as impacts.

20. Thus it is inherent in Zager that faults fall into different classes. If all faults were regarded equally, there would be no utility in determining the significance of the condition-changing event(s). Zager explicitly differentiates faults into root and non-root faults.

21. At column 11, lines 54-57, Zager teaches:

> An impact is the description of a disruption in service for some portion or user A of the external system owing to a correlated disruption in service of some portion B.

22. It is inherent that a disruption in service has a duration and that a time to recovery from a fault (or equivalently, to repair a failure) is a parameter of a disruption in service. For example, the concept of announcing outages in a service, whether computer-implemented or real world, with an accompanying duration the outage is expected to last is well known.

23. Further, the concepts of “mean time between failure” and “mean time to repair” are metrics well known in the art for expressing reliability, and refer to a failure rate and a repair time, respectively. See Newton’s Telecom Dictionary.

24. Further, it is well known in the art of modeling and simulation that a model for a system must be complex enough to accurately represent the system being modeled. See Banks, section 1.3.1. Special attention is paid to events, defined as “an occurrence that changes the state of the system”. A failure and a recovery from a failure are clearly occurrences that change the state of a network of computers and are inherently modeled in a reliability model of a network.

25. Another concept well known in the art of modeling is the importance of activity and delay in the model. See Banks, section 1.3.6. In a system where activity and delay directly impact the effectiveness of the system, a model of such a system inherently models the periods of activity and delay. Failure to do so would render the model inaccurate and useless. In the example of a computer network, it is well known that failures in the network components, the duration that they are in a state of failure, and their recovery from failure are events related to activity and delay which are critical to determining the reliability of the computer network.

26. Additionally, at column 13, lines 44-50, Zager teaches:

> The desired result of monitoring is:
> To recognize the onset or termination (hereinafter sometimes termed the "offset") of faults by interpreting the messages a device or application has emitted; and
> To detect the onset or offset of anomalous behavior by performing quantitative analytic tests against performance metrics

27. Thus Zager clearly teaches recognizing the time of fault and the time of recovery from fault (equivalently failure and repair) in a system that provides reliability data.

28. Lastly, at column 11, lines 9-16, Zager teaches:

> Services, therefore, create a dimension of impact analysis from differing perspectives. For example, suppose it is desired to know how reliable database service is for some given set of disparate users. Categorization by service allows the model to cut simply across different database service providers and to provide an aggregation of both fault and impact data for that service.

29. Thus Zager clearly teaches an aggregation of reliability data. As explained above, Zager teaches monitoring faults and disruptions in service. It is inherent that the aggregated reliability data are the result of the faults and disruptions in service as providing such is a goal of Zager's invention and the motivation for the modeling system. It is inherent that a reliability model for a computer network models the activity and delays in the computer network, functionally equivalent to failures and recovery from failures. It is well known to express reliability as mean time between failure and mean time to repair. Zager does teach "an aggregate failure rate and aggregate repair time for each said classes of failures in the form of aggregate fault and impact data" and therefore the previous rejection of claim 1 under 35 U.S.C. § 102 based on the Zager reference is maintained.

30. Regarding Applicant's arguments regarding claim 2, Applicant argued:

> However, Zager fails to teach the software availability model as discussed with reference to claim 1, and further, there is no teaching that the platform parameters <u>define platform problems causing failures and affecting recovery times related to the platform problems</u>.

31. The Examiner respectfully traverses Applicant's argument based on the following remarks.

32. Zager's teaching of the software availability model of claim 1 has been discussed above. Regarding platform problems causing failures and affecting recovery times related to the platform problems, at column 12, lines 23-33, Zager teaches:

> During the course of normal operation, managed resources (the elements or components, including both hardware and software, of the external system) suffer faults and performance degradations. New managed resources join the external system, others leave the system, and others change their configuration relative either to themselves (e.g., an equipment upgrade) or to other resources in the external system. All these systemic occurrences will be termed incidents.

33. Since it would be impossible to determine a performance degradation has occurred without a predefined notion of normal performance, it is inherent that Zager teaches defining a problem causing a fault. Additionally, at column 11, lines 23-36, Zager teaches:

> Many resources (parts of the external system) report their faults, whether through spontaneously recording to a log, emission of traps of the type provided for by the Simple Network Management Protocol ("SNMP"), or in response to an active request for information on performance. Some resources, however, are not constructed with the ability to report their own state spontaneously, and are not instrumented to respond to a polling request or the like; this necessitates fault inference. What this means is simply that the model infers that a fault exists. Passive monitoring techniques such as pinging a task or device, give the barest of information, essentially only whether the task or device is still running in the system.

34. Pinging a task or device to determine if it is still running in the system makes inherent that it has been defined that the task or device should be running in the system, which reiterates Zager's teaching of defining a problem that causes a fault. If a task or device that is expected to be running fails to respond to a ping, it is known that a problem has occurred.

35. Regarding recovery times, it has been explained above that disruptions in service have respective durations, the concept of mean time to repair is well known, the importance of accurately modeling activity and delay in a model is well known, and it

has been established that Zager models reliability in a computer network. Thus it is inherent in the reliability model of Zager that a recovery time is known when a given fault has been detected.

Regarding claim 16, Applicant argued:

> Zager does not teach that "impacts" include a failure rate for a particular software error and does not teach determining the recovery rate from such software error. Impacts as used by Zager has to do with one service being affected by the degradation or unavailability of another service or component (e.g., cannot access a database when a router is out and the like).

36. The Examiner respectfully traverses Applicant's arguments based on the following remarks.

37. The issue of failure rates and recovery rates has been discussed above in paragraphs 14-29. To summarize, Zager clearly teaches faults and recovery from faults, functionally equivalent to a failure and a repair in response to a failure. Zager clearly teaches that faults fall into different classes. It is inherent that disruptions in service have a duration, the concept of mean time between failure and mean time to repair are well known, the importance of modeling activity and delay are well known, and Zager clearly teaches modeling the reliability of a computer network. Zager clearly teaches an aggregation of reliability data. Therefore it is inherent in Zager's reliability model to include a failure rate and recovery rate for a particular error, whether it be software or otherwise. Omitting these details would render the Zager's reliability model at best inaccurate but more typically useless.

35 U.S.C. § 103

38. Regarding Applicant's response to the rejections under 35 U.S.C. § 103 of claims 6-15, 17, and 18, Applicant argued:

> Categorization does not require including failure rates for software components on a node and aggregated values of such rates in a software availability model and also, does not teach including repair times for the software components as an aggregated value. As discussed in Applicant's specification, the failure rates and repair times may vary widely and aggregation allows a single value to be used in a model. Zager fails to teach the software availability model of claim 6, and at the portion cited in the Office Action, Zager teaches aggregating fault and impact data for a service but, again, the impact data is different than called failure rates and repair times, and how reliable a database service is for a "given set of disparate users" does not teach the software reliability model of claim 6.

39. The issue of failure rates and recovery rates has been discussed above in paragraphs 14-29. To summarize, Zager clearly teaches faults and recovery from

faults, functionally equivalent to a failure and a repair in response to a failure. Zager clearly teaches that faults fall into different classes. It is inherent that disruptions in service have a duration, the concept of mean time between failure and mean time to repair are well known, the importance of modeling activity and delay are well known, and Zager clearly teaches modeling the reliability of a computer network. Zager clearly teaches an aggregation of reliability data. Therefore it is inherent in Zager's reliability model to include a failure rate and recovery rate for a particular error, whether it be software or otherwise. Omitting these details would render the Zager's reliability model at best inaccurate but more typically useless.

40. Applicant reiterates this argument in reference to claims 8 and 17, however the Examiner respectfully traverses these arguments as above in reference to claim 6.

Claim Rejections - 35 USC § 112

41. The following is a quotation of the first paragraph of 35 U.S.C. § 112:

> **The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.**

42. Claims 8-15 and 18 are rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. The terms "warm recoverable error" and "non-warm recoverable error" are not defined by the disclosure. On page 1 of Applicant's response, Applicant argues that these terms have accepted meaning in the technical arts, however a search of

"Dictionary of Computer Science, Engineering, and Technology", "IBM Dictionary of Computing", "IEEE 100, The Authoritative Dictionary of IEEE Standards Terms", "Microsoft Computer Dictionary", internet search engine Google, CiteSeer Scientific Literature Digital Library, and Safari Technical Books text search (all attached) has failed to disclose any reference to the terms. It is therefore the conclusion of the Examiner that these terms do not have an accepted meaning in the technical arts. Further, since they are not defined by the disclosure through neither explicit definition nor clear example, an artisan of ordinary skill would be unable to build or use the invention recited by claims 8-15 and 18.

43. The following is a quotation of the second paragraph of 35 U.S.C. §112:

> **The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.**

44. Claims 8-15 and 17-18 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

45. Regarding claims 8 and 18, it is unclear what is meant by "warm recoverable errors". It is unclear what is meant by "warm recoverable error state parameters". It is unclear what is meant by "non-warm recoverable errors". It is unclear what is meant by "non-warm recoverable error state parameters". As set forth above, the terms "warm recoverable error" and "non-warm recoverable error" are ambiguously defined by the specification and are not well known in the art. As a result, it is not possible to determine the metes and bounds of claims 8 or 18.

46. Further regarding claims 8 and 18, it is unclear what is meant by "warm recoverable error state parameters" and "non-warm recoverable error state parameters" as they relate to "warm recoverable errors" and "non-warm recoverable errors".

47. Claims 9-12 recite the terms "warm recoverable software errors" and "non-warm recoverable software errors". It is unclear how these terms differ from "warm recoverable errors" and "non-warm recoverable errors" and are therefore rejected for the same reasons set forth above in the rejection of claim 8 under 35 U.S.C. § 112, second paragraph. The Examiner observes that the teachings of Applicant's specification at page 24, appear to teach away from the concept of a "non-warm recoverable <u>software</u> error":

> Thus, software availability may be impacted by errors that result in recovery actions in the applications, or warm recoverable, or errors that result in recovery actions on the node or cluster, or non-warm recoverable.

48. However, page 7 teaches that

> Operating system 104 and software application 108 can be considered the software components of node 102. Repairs to software components may include restarting the application, rebooting node 102, and other activities that should not necessitate hardware fixes or repairs.

49. It is therefore the Examiner's interpretation, explained in the previous Office Action and asserted above in Response to Arguments, that a "non-warm recoverable error" refers to a hardware error, however that interpretation renders the meaning of a "non-warm recoverable <u>software</u> error" vague and indefinite beyond its similarity to "non-warm recoverable errors".

50. The term "a fraction of recovery failures" in claims 9, 10, 11 and 17 is a relative term which renders the claim indefinite. The term "a fraction of recovery failures" is not defined by the claim, the specification does not provide a standard for ascertaining the

requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention. Examiner observes that all rational numbers are defined by their ability to be represented as a fraction.

51. Any claim rejected but not specifically mentioned is rejected by virtue of its dependence.

Claim Interpretation

52. Regarding claim 8, the term "warm recoverable errors" is interpreted as "software error" as set forth in the previous Office Action.

53. The term "non-warm recoverable errors" is interpreted as "hardware error" as set forth in the previous Office Action.

54. The term "warm recoverable error state parameters" is interpreted as "software error rate and time to recover" as set forth in the previous Office Action.

55. The term "non-warm recoverable error state parameters" is interpreted as "hardware error rate and time to recover" as set forth in the previous Office Action.

56. Regarding claim 9, 10 and 17 the term "a fraction of recovery failures" is interpreted as "a percentage of recovery attempts that failed" as set forth in the previous Office Action.

Claim Rejections - 35 USC § 102

57. The following is a quotation of the appropriate paragraphs of 35 U.S.C. §102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

58. Claims 1-5, 16, and 19 are rejected under 35 U.S.C. §102(e) as being anticipated by Zager et al. (US 006393386 B1)

59. Regarding claim 1, Zager et al. discloses a method of modeling a complex system, such as a distributed computing ensemble, wherein

platforms with at least one software component are modeled (column 3, lines 4-11; column 8, lines 35-38; column 8, lines 42-43),

failures are modeled (column 5, line 63 – column 6, line 7; column 11, lines 17-23),

recovery from failures are modeled (column 11, lines 18-23),

the failures belong to different classes (column 7, lines 4-18) including root causes, non-root causes, and performance degradation failures,

a platform is modeled (column 5, lines 23-28; column 8, lines 36-38; column 8, lines 42-43), and

the model includes an aggregate failure rate and aggregate repair time for each of said classes of failures in the form of aggregate fault and impact data (column 11, lines 10-16).

60. Regarding claim 2, Zager et al. discloses modeling different platforms (column 3, lines 4-11; column 8, lines 35-38; column 8, lines 42-43). It is deemed inherent that platforms parameters are required to model a platform.

61. Regarding claim 3, Zager et al. discloses modeling hardware components (column 3, lines 4-11; column 3, lines 17-21; column 3, lines 48-54).

62. Regarding claim 4, it is deemed inherent that the mean time to repair includes time to detect and identify an error. The time to detect and identify an error has an impact on the availability of the network resources. It is a stated goal of Zager et al. to monitor and model the state of network resources as well as the impacts of events, which are faults and recovery from faults, therein (column 2, line 64 – column 3, line 3).

63. Regarding claim 5, Zager et al. discloses that the platforms are nodes in a network (column 8, lines 35-50).

64. Regarding claim 16, Zager et al. discloses a method of modeling a complex system, such as a distributed computing ensemble, wherein

events such as failures and performance degradations are represented in the model (column 5, lines 63-64; column 11, lines 18-22),

a recoverable state, represented as root or non-root, is determined for said event (column 7, lines 4-18; column 31, line 54 – column 32, line 40),

a failure rate and recovery rate is determined for said event (column 3, lines 36-47), and

event data is incorporated into the recoverable state data (column 3, lines 4-11; column 3, lines 36-47).

65. Regarding claim 19, Zager et al. discloses a method of modeling a complex system, such as a distributed computing ensemble, wherein

the model is a software model (column 3, lines 28-31),

events such as failures and performance degradations are represented in the model (column 5, lines 63-64; column 11, lines 18-22),

a recoverable state, represented as root or non-root, is determined for said event (column 7, lines 4-18; column 31, line 54 – column 32, line 40),

a failure rate and recovery rate is determined for said event (column 3, lines 36-47), and

event data is incorporated into the recoverable state data (column 3, lines 4-11; column 3, lines 36-47).

Claim Rejections - 35 USC § 103

66. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> **(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.**

67. Claims 6-7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zager et al. (US 006393386 B1) in view of Cristian.

Regarding claim 6, Zager et al. discloses a method and system for modeling a complex system such as a distributed computing ensemble wherein

at least one node is modeled (column 8, lines 2-7; column 8, lines 42-43; column 8, lines 44-50),

services, both computational and functional, represent software (column 9, line 55 – column 10, line 5), and

a software availability model (column 11, lines 10-16) includes an aggregated failure rate and an aggregated repair time for each software component (column 11, lines 10-16) as taught by categorization of reliability according to service.

68. Zager et al. does not disclose modeling errors and recoveries at a level of detail including reboot times.

69. Cristian teaches a framework for fault-tolerant distributed computing systems including various hardware and software failures exhibited by such systems. In particular, Cristian teaches the importance of early timing failures and late timing failures, grouped as performance failures (page 58, column 2, line 60 – column 3, line 7). Cristian also teaches the importance of crash failures, which require that a server restarts with potential loss of state or data (page 58, column 3, lines 13-35). Servers can be implemented in hardware or software (page 57, column 3, lines 23-34).

70. The combination of a crash error in a hardware server, which requires the server to restart, and performance failure data defining the timing window in which a server should perform a restart action constitute a reboot time. Thus, Cristian teaches the use of a reboot time to detect errors in a distributed computing system.

71. It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to incorporate the reboot time parameters in the node model of an availability model because a node is not available during the reboot sequence and monitoring reboot times is a simple and known method of checking for unusual operation during the reboot process.

72. The combination of the reboot time parameter to the node model of Zager et al. would be easily achieved by marking the time when a service begins a reboot sequence and subsequently polling the service at the beginning and end of the timing window. In effect, checking for early timing failures and late timing failures. Incorporating such a feature into the invention of Zager et al. would produce a network model with the ability to detect a wider range of typical errors.

73. Regarding claim 7, Zager et al. discloses that managed resources, including hardware components, may suffer faults or performance degradations (column 12, lines 23-27) that are of interest to the model (column 11, lines 42-46). It is deemed inherent that these hardware components are modeled corresponding to the node models.

74. Claims 8-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zager et al. (US 006393386 B1) in view of Cristian.

75. Regarding claim 8, Zager et al. discloses a system for modeling complex networks, such as a distributed computing ensemble, and predicting the impacts of faults therein where hardware and software resources are modeled (column 3, lines 4-11). Zager et al. also models the recovery from faults. Faults and recoveries from faults are modeled as events (column 11, lines 18-23). Zager et al. discloses determining whether an error is a root cause or non-root cause error (column 7, lines 4-18; column 31, line 54 – column 32, line 40).

76. Zager et al. does not disclose modeling errors and recoveries at a level of detail including warm recoverable errors or non-warm recoverable errors. Zager et al. does

not disclose the use of parameters related to warm recoverable errors or non-warm recoverable errors.

77. Cristian teaches a framework for fault-tolerant distributed computing systems including various hardware and software failures exhibited by such systems. In particular, Cristian teaches that servers may be implemented in hardware or software (page 57, column 3, lines 23-34) and the concept of warm recoverable errors (partial-amnesia crash and pause-crash) as well as non-warm recoverable errors (amnesia crash) (page 58, column 3, lines 13-34). Cristian also teaches the importance of performance failures, specifically early timing failures and late timing failures (page 58, column 2, line 60 – column 3, line 7), which teach the recovery rates for warm and non-warm recoverable errors when applied to the recovery procedures following crash failures.

78. It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to incorporate these failures and corresponding data into the system for modeling complex networks, such as distributed computing ensembles, because it was known that different errors had different characteristics and recovery procedures and a goal of modeling a complex system is to represent its performance as accurately as possible.

79. The combination of these errors with the invention of Zager, et al. would be easily achieved by incorporating them as additional data in the root cause determination or performance degradation failure states, as well as the parameters and state data necessary to model the warm recoverable errors and non-warm recoverable errors.

Incorporating such a feature into the invention of Zager et al. would produce a network model with higher fidelity regarding the actual complex network.

80. Regarding claim 9, Zager et al. models both hardware and software components (column 3, lines 4-11). Both hardware and software components experience faults (column 12, lines 23-32). Thus, Zager et al. teaches recovery attempts for software errors that fail 0% of the time.

81. Regarding claim 10, Zager et al. teaches recovery attempts for software errors that fail 0% of the time as rejected for claim 9. Zager et al. also teaches the determination of an event as root cause or non-root cause (column 7, lines 4-18; column 31, line 54 – column 32, line 40). The percentage of recoveries that fail is in intrinsic property of the recovery attempts and therefore does not require a dedicated operation by the invention of Zager et al. Thus Zager et al. teaches generating the percentage of recovery attempts that fail in the same step as generating the state parameters.

82. Regarding claim 11, Zager et al. teaches modeling events, which is a fault or the recovery from a fault (column 11, lines 18-23). Zager et al. models both hardware and software components (column 3, lines 4-11). Both hardware and software components experience faults (column 12, lines 23-32). Thus, Zager et al. teaches recovery attempts for hardware errors that fail 0% of the time.

83. Regarding claim 12, Zager et al. teaches recovery attempts for hardware errors that fail 0% of the time as rejected for claim 11. Zager et al. also teaches the determination of an event as root cause or non-root cause (column 7, lines 4-18; column 31, line 54 – column 32, line 40). The percentage of recoveries that fail is in

intrinsic property of the recovery attempts and therefore does not require a dedicated operation by the invention of Zager et al. Thus Zager et al. teaches generating the percentage of recovery attempts that fail in the same step as generating the state parameters.

84. Regarding claim 13, node recovery parameters are deemed synonymous with non-warm recoverable errors, referred to by Cristian as an amnesia crash (page 58, column 3, lines 13-34), in combination with performance errors (page 58, column 2, line 60 – column 3, line 7), which provide a timing window by which recovery from a non-warm recoverable error can be measured. The motivation for this combination is given in the rejection of claim 8 above.

85. Regarding claim 14, node recovery parameters are deemed synonymous with non-warm recoverable errors, referred to by Cristian as an amnesia crash (page 58, column 3, lines 13-34), in combination with performance errors (page 58, column 2, line 60 – column 3, line 7), which provide a timing window by which recovery from a non-warm recoverable error can be measured. Cristian teaches that servers may be hardware or software (page 57, column 3, lines 23-34). The recovery from an amnesia crash experienced by a server implemented in hardware is deemed synonymous with a reboot procedure. The performance data related to the reboot procedure constitutes node reboot parameters. The motivation for this combination is given in the rejection of claim 8 above.

86. Regarding claim 15, Zager et al. teaches that various software components acquire information relating to the operation of the external system and report changes of state in the modeled components to the model (column 6, line 62 – column 7, line 1).

The external system is the complex network of computing devices (column 5, lines 48-50). It is deemed inherent that when the network or a portion of the network reboots, the software components that acquire information relating to the operation of the external system detect this change of state and report the relevant parameters to the model.

87. Claim 17 is rejected under 35 U.S.C. §103(a) as being unpatentable over Zager et al. (US 006393386 B1) as applied to claim 16 above, and further in view of Cristian.

88. Zager et al. does not explicitly teach modeling recovery failures except as inherent in the mean time to recover statistic.

89. Cristian teaches a framework for fault-tolerant distributed computing systems including various hardware and software failures exhibited by such systems. In particular, Cristian teaches response failures which correspond to situations where the server's state transition is incorrect (page 58, column 3, lines 7-13). This type of response failure is deemed synonymous with a recovery failure that is intended to model misdiagnosis of the failure, a corruption in the checkpoint stored for the application, and miscellaneous failures to restart (instant application, paragraph 0031, lines 6-9).

90. It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to make this combination because metrics for the reliability of distributed computing including software, hardware, and recovery failures were known in the art. In building a model for such a system, it would have been obvious to incorporate the established metrics for such a system in the model.

91. The combination of response failures from Cristian with the method of modeling a complex system of Zager et al. would be easily produced by including a response failure as an additional type of event. Zager et al. currently models faults and recovery from those faults. The combination would allow for recovery attempts that fail to correct the fault and subsequently the object experiencing a fault would not transition to a recovered state. Such a combination would produce an availability model that represents an actual computer network with greater fidelity.

92. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Zager et al. (US 006393386 B1) in view of Cristian.

93. Zager et al. discloses a system for modeling complex networks, such as a distributed computing ensemble, and predicting the impacts of faults therein where hardware and software resources are modeled (column 3, lines 4-11). Zager et al. also models the recovery from faults (column 11, lines 18-23). Zager et al. teaches that the model is a software model (column 3, lines 28-31).

94. Zager et al. does not disclose modeling errors and recoveries at a level of detail including warm recoverable errors or non-warm recoverable errors. Zager et al. does not disclose the use of parameters related to warm recoverable errors or non-warm recoverable errors.

95. Cristian teaches a framework for fault-tolerant distributed computing systems including various hardware and software failures exhibited by such systems. In particular, Cristian teaches that servers may be implemented in hardware or software (page 57, column 3, lines 23-34) and the concept of warm recoverable errors (partial-

amnesia crash and pause-crash) as well as non-warm recoverable errors (amnesia crash) (page 58, column 3, lines 13-34). Cristian also teaches the importance of performance failures, specifically early timing failures and late timing failures (page 58, column 2, line 60 – column 3, line 7), which teach the recovery rates for warm and non-warm recoverable errors when applied to the recovery procedures following crash failures.

96. It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to incorporate these failures and corresponding data into the system for modeling complex networks, such as distributed computing ensembles, because it was known that different errors had different characteristics and recovery procedures and a goal of modeling a complex system is to represent its performance as accurately as possible.

97. The combination of these errors with the invention of Zager, et al. would be easily achieved by incorporating them as additional data in the root cause determination or performance degradation failure states, as well as the parameters and state data necessary to model the warm recoverable errors and non-warm recoverable errors. Incorporating such a feature into the invention of Zager et al. would produce a network model with higher fidelity regarding the actual complex network.

Conclusion

References relied upon to establish the state of the art have been cited on form PTO-892.

Claims 1-19 have been presented for reconsideration in view of Applicant's amended claim language and arguments. Claims 1-19 have been rejected.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office Action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the data of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin J Teska can be reached on (571) 272-3716. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Proctor
Examiner
Art Unit 2123

Jason Proctor
jsp

KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER